

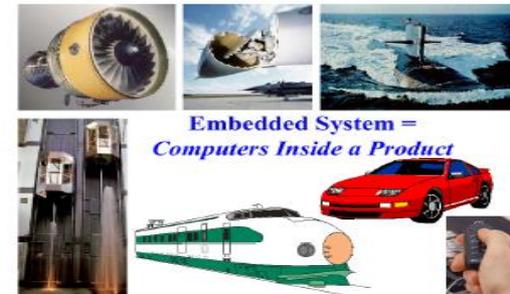
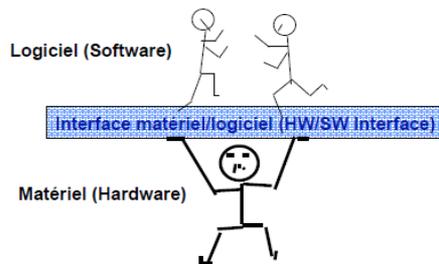
Master 1 : Systèmes embarqués et Mobilité

Matière : Processeurs embarqués

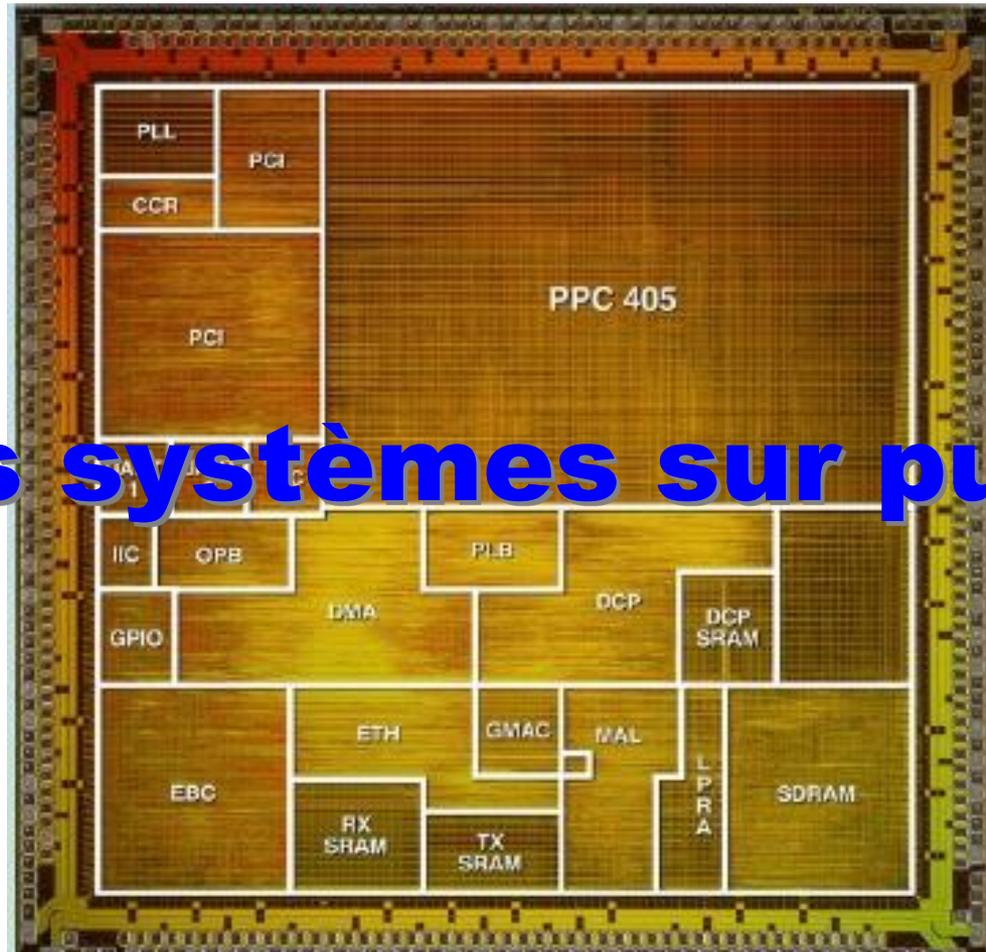
Cours 0 : Introduction aux systèmes embarqués (SE) - 2

Année 2022 / 2023

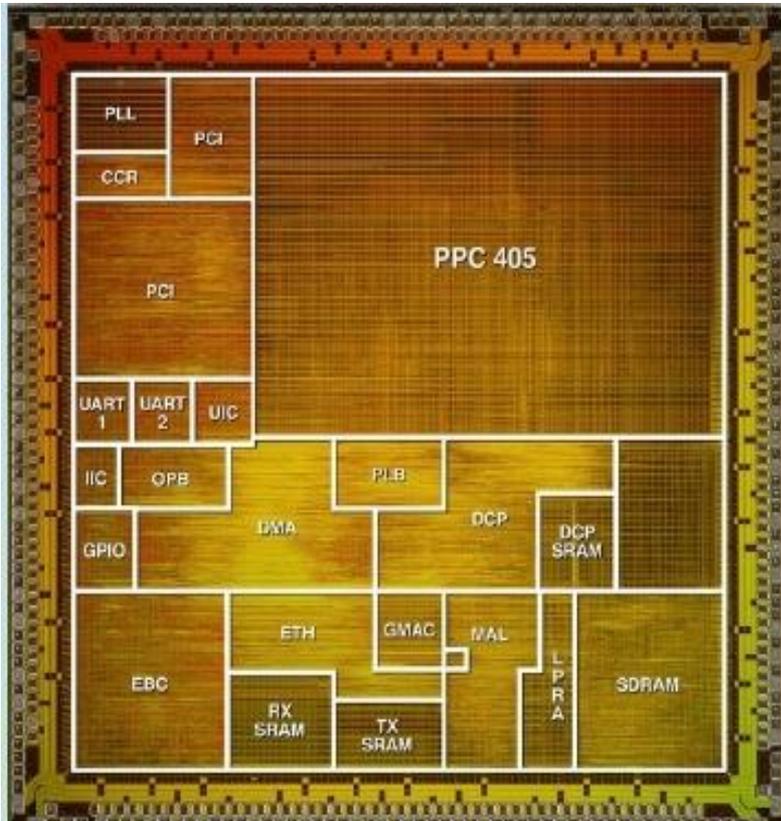
Pr R. BOUDOUR



Les systèmes sur puce



Que signifie un SoC ?

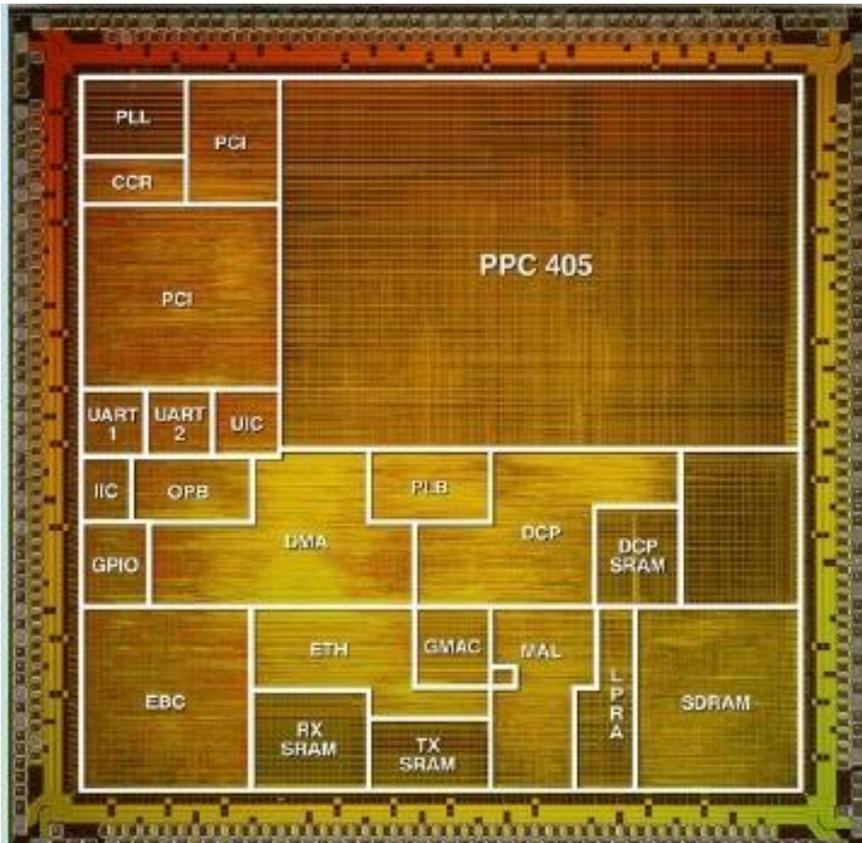


- **SoC** = System on Chip =
Système sur Puce

- **Définition**

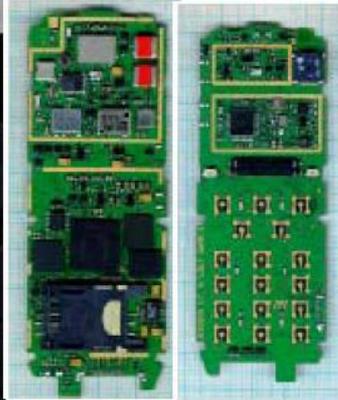
Circuit intégré complexe qui intègre un grand nombre d'éléments fonctionnels hétérogènes sur une seule puce pour constituer un produit complet.

Que signifie un SoC ?

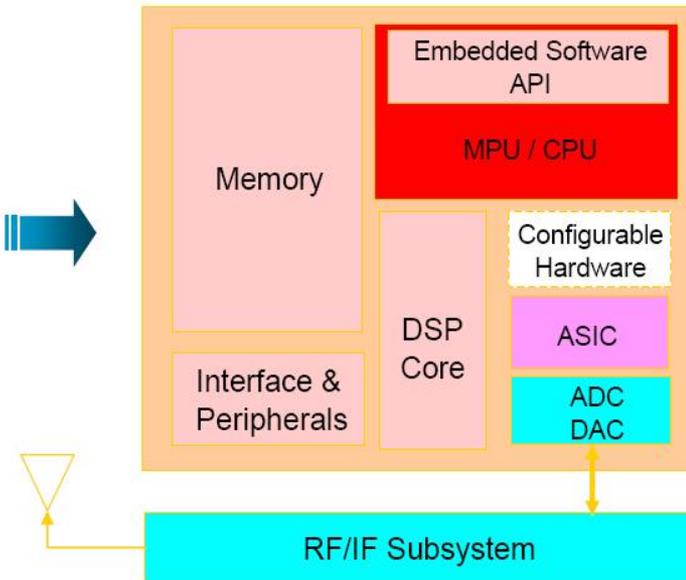


- Système sur Puce « SoC » contient :
 - IPs réutilisables
 - Processeurs embarqués
 - Mémoires embarquées
 - Interfaces standard (USB, PCI, Ethernet)
 - Logiciels embarqués
 - Modules Signaux mixtes (CAN/CNA)
 - Logique programmable FPGA > 500k portes
- En pratique
 - Un super ensemble d'architectures qui sera adapté par l'utilisateur
 - Spécifier l'architecture selon les besoins en enlevant et rajoutant des composants adaptés.
- Le développement initial est coûteux mais les systèmes dérivés seront moins chers

Que signifie un SoC ?



System-on-Board
(SoB)



System-on-Chip
(SoC)

Applications SoC

- Communication
 - Téléphone cellulaire
 - Réseaux

- Ordinateur : PC/Station de travail

- Consommation grand public
 - Appareil photo numérique
 - Station de jeux
 - etc.

Avantages des SoC

- ❑ Réduction du coût global du système
 - ❑ SoC réutilisable pour une famille d'applications
 - ❑ Utiliser des composants existants (IPs)
 - ❑ Utiliser tout ce qui est opensource

- ❑ Augmenter les performances en réalisant certaines choses sur mesure

- ❑ Réduction de l'encombrement et de la consommation en énergie

Comment surmonter la complexité ?

- ❑ **Systemes complexes :**
 - mCs, DSPs, Hw/Sw, RTOS's, Bus sur puce, IP digital/analogique, Milliard de transistors, coûts du masque, etc.

Solutions

- ❑ Utiliser des plateformes connues
- ❑ Utiliser des IPs testés et validés
- ❑ Partitionner selon les fonctionnalités du SoC et sous forme de Hw et de Sw
- ❑ Modéliser le SoC à différents niveaux d'abstractions

Notion d'IP (Intellectual Property)

□ Définition d'IP cores :

- Blocs fonctionnels complexes réutilisables, testés et vérifiés, et protégés par un copyright (droits d'utilisation, licences).
- Les IPs sont appelés aussi : méga-fonctions, macro-blocs, composants réutilisables.

On peut se procurer des noyaux **sous trois formes** :

- **"soft"** - un modèle, une description de haut niveau en VHDL, Verilog ou autre, avec typiquement un grand degré d'indépendance vis-à-vis de la technologie-cible (FPGAs, VLSI dédié ou semi-dédié, etc.)
- **"firm"** - un modèle au niveau des portes logiques ou un modèle RTL plus difficile à modifier et probablement optimisé pour une technologie-cible particulière.
- **"hard"** - un modèle physique placé et routé de façon optimale, impossible à modifier et totalement dépendant de la technologie-cible

Obtention des IPs

□ Quelques liens :

- [opencores.org](http://www.opencores.org) (www.opencores.org) offre entre autres un module d'interface Ethernet, un processeur compatible 8051 et un contrôleur VGA / LCD.
- "Arithmetic Cores" , des noyaux à vocation arithmétique incluant un registre à barillet, une unité de calcul en point flottant, etc.

www.geocities.com/SiliconValley/Pines/6639/ip/arithcores.html

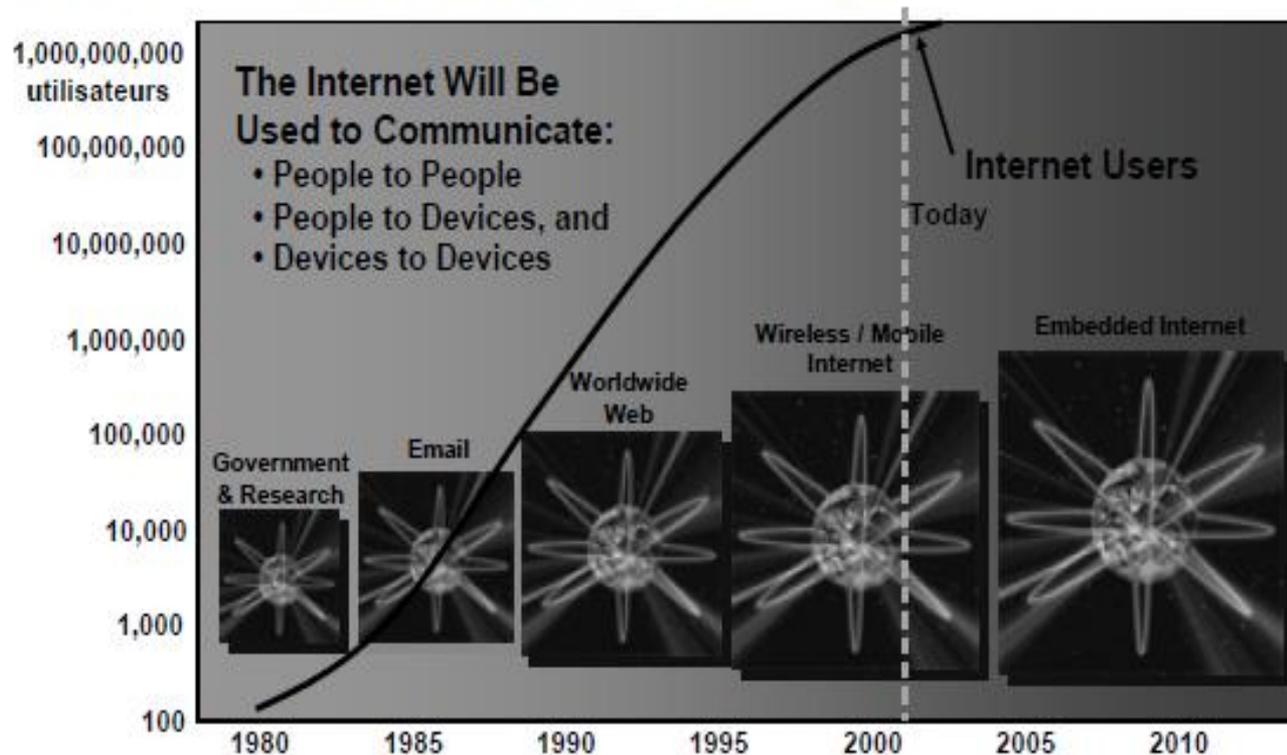
- "The Free-IP Project", des noyaux pour FPGA et pour VLSI dédiés incluant un processeur 6502, un décrypteur DES, etc. (www.free-ip.com)

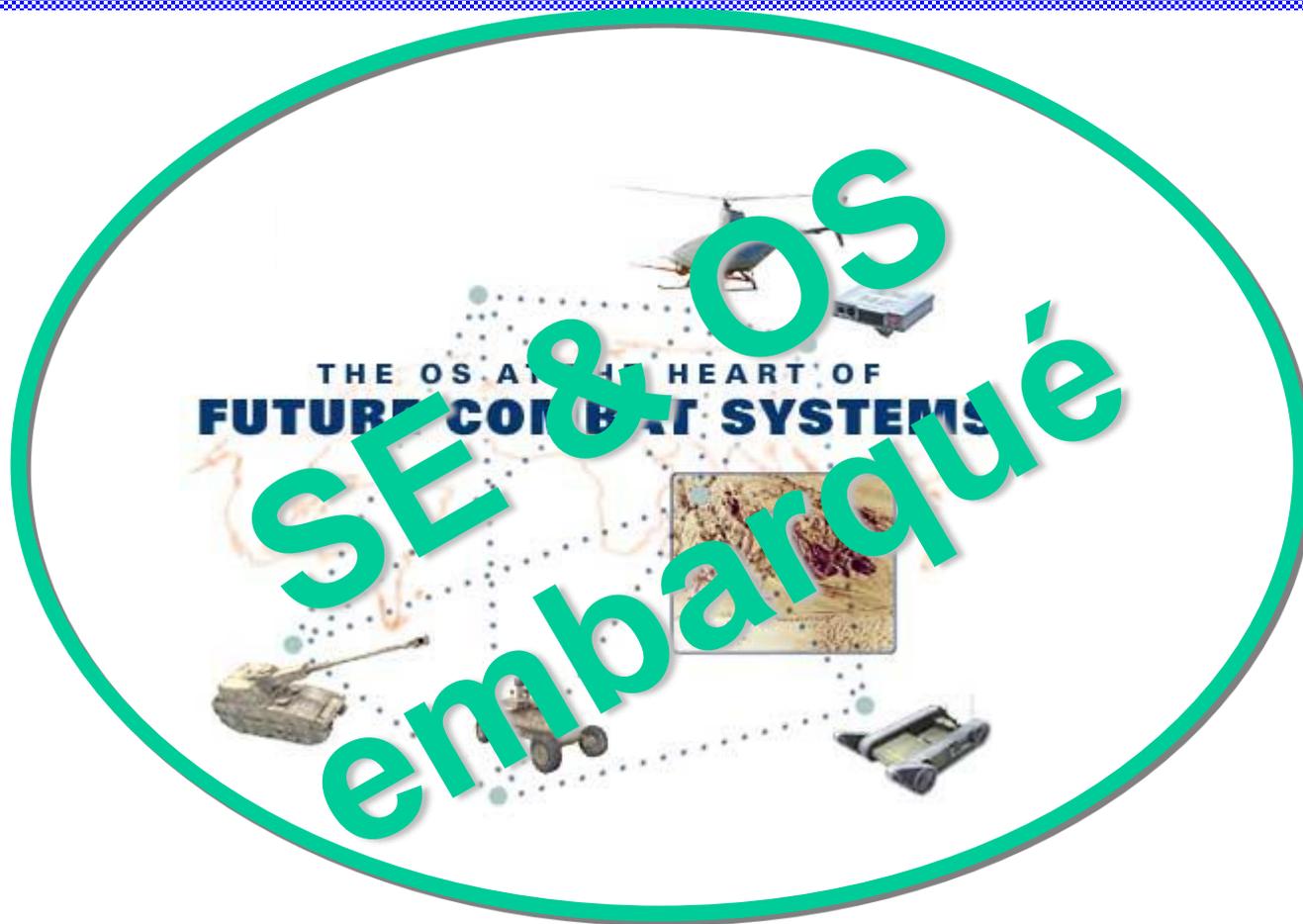
Connectivité IP & SE

- ❑ La connectivité Internet permet de raccorder tout système électronique (système embarqué) au réseau Internet. On parle aussi de connectivité IP (*Internet Protocol*).
- ❑ Ajouter une connectivité IP à un système électronique permet de le contrôler à distance de n'importe où dans le monde :
 - par une application réseau.
 - plus simplement par le « web » en utilisant un navigateur Internet (Netscape, Internet Explorer, ...).

Connectivité IP & SE

Explosion du marché de l'Internet embarqué

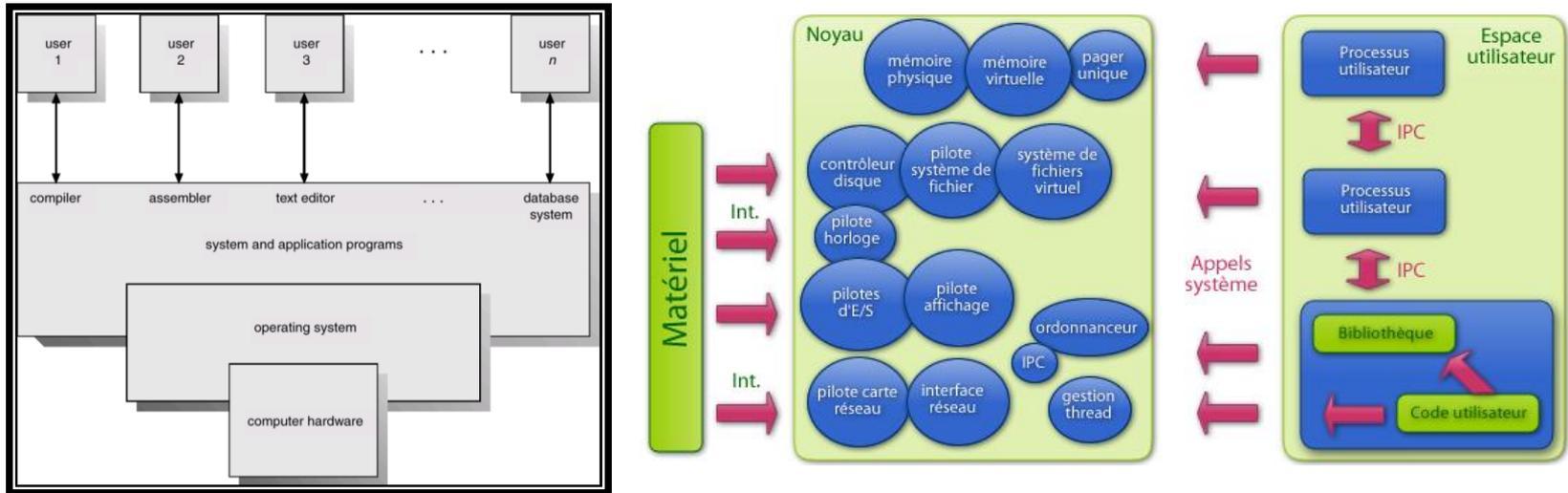




SE & OS embarqué

- ❑ Le système d'exploitation est un programme (couche optionnelle) qui contrôle l'exécution des programmes d'application et qui agit comme une interface entre l'utilisateur et le matériel. Il a pour objectif de faciliter l'exploitation du matériel et son partage d'une façon efficace.
- ❑ Le matériel fournit des ressources telles que les unités de traitement (CPU, co-processeurs, processeurs graphiques, etc.), les unités de stockage (mémoire, cache, disque, etc.) et les E/S (terminaux, écran, etc.). L'OS facilite l'accès des applications à ces ressources.

SE & OS embarqué



L'OS permet par exemple :

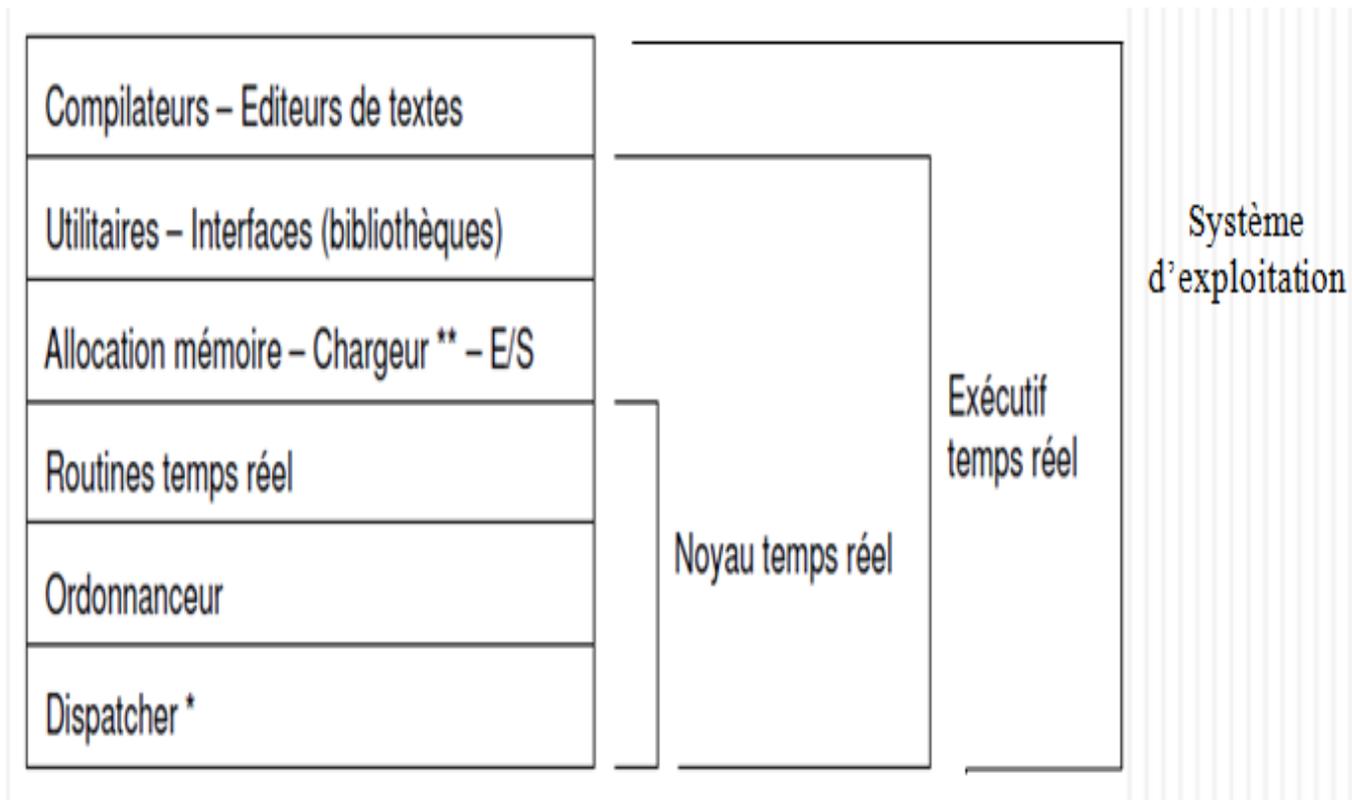
- La gestion des fichiers, les accès au disque
- L'interfaçage avec le clavier et l'écran
- L'ordonnancement de plusieurs programmes (ou de plusieurs threads) pour l'exécution
- D'être appelé à partir d'un programme utilisateur

SE & OS embarqué

□ RTOS : Real Time Operating System

- RTOS est un système d'exploitation développé pour des applications en temps réel et typiquement employé pour des applications embarquées sur des systèmes spécifiques avec un ensemble de fonctions limitées déterminées par la conception du matériel.
- Il constitue un élément important des systèmes embarqués aujourd'hui
- Il est caractérisé par :
 - Un comportement déterministe : les opérations sont effectuées à des temps fixes et prédéterminés ou bien à des intervalles de temps prédéterminés.
 - Le temps de réponse : Il porte sur le temps nécessaire au système d'exploitation afin de servir une interruption après que le signal acknowledgment (acquiescement) ait été émis.
 - La contrôlabilité par l'utilisateur
 - La fiabilité, etc.

SE & OS embarqué



Exemples OS

- ❑ OS embarqués
 - uClinux
 - Windows CE
- ❑ OS temps réels
 - VxWorks
 - Linux + patch PREEMPT-RT
 - Xenomai
- ❑ OS spécialisés
 - Android
 - BlackberryOS
 - iOS
 - OSEK/VDX

Choix de processeur ?

❑ Lesquels ?

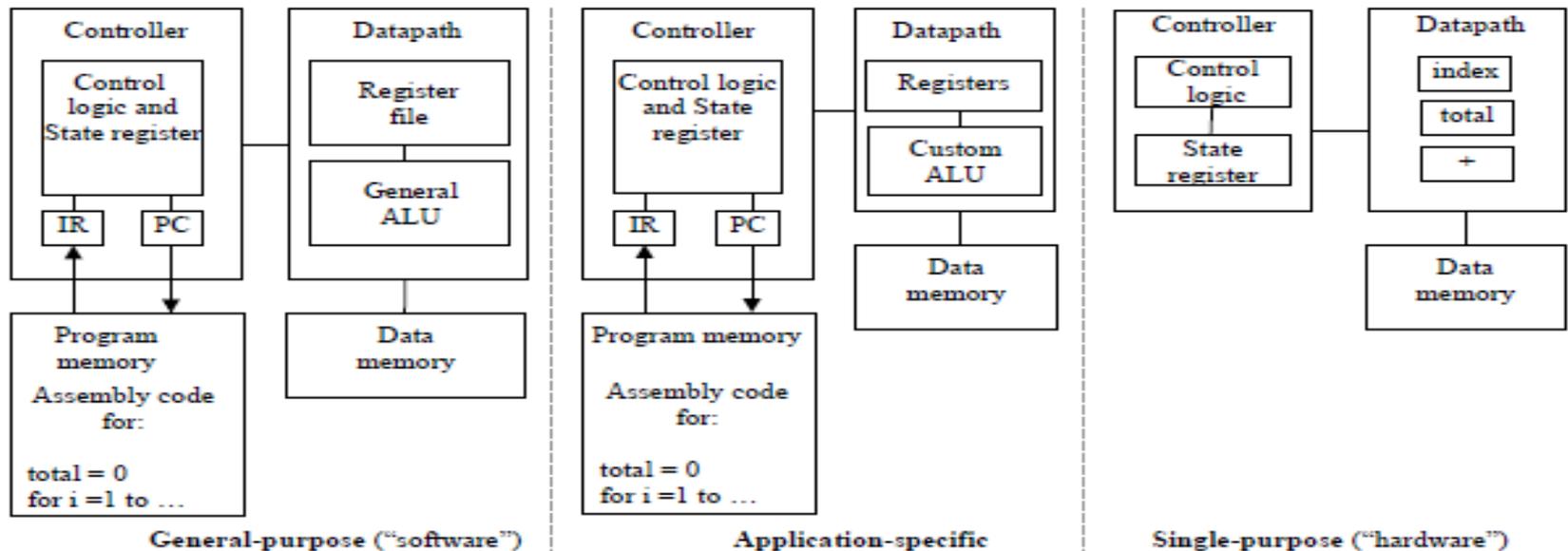
- généralistes
- embarqués/enfouis haute performance

❑ Convergence

- Besoins en performance
- Similarités croissantes des architectures
- Contraintes de l'embarqué :
 - Consommation
 - Taille de code
 - Coût
 - Temps-réel

Choix de processeur ?

- ❑ Plusieurs possibilités pour fournir un service :
 - Processeur généraliste
 - Spécifique pour une application
 - Dédié



Choix de processeur ?

- ❑ Pourquoi faire appel à une unité programmable ?
- ❑ Pourquoi ne pas utiliser une solution totalement en matériel (Hardwired) ?
 - L'utilisation du logiciel permet de spécialiser plus rapidement et plus facilement un SE
 - Mise au point du système facile et rapide
 - Solution de faible coût
 - Dans la plupart des SE, le SW domine le coût du SE. La part du logiciel dans les SE ne cesse d'augmenter

Choix du processeur ?

- Le choix du bon processeur pour l'usage dans une application embarquée exige une évaluation soignée d'une variété de facteurs :
 - Performances - combien de Millions d'Instructions Par Seconde (MIPS) sont nécessaires pour accomplir une tâche dans le temps accordé ?
 - Environnement de fonctionnement - Dans quelle température ambiante le système fonctionnera t-il ?
 - Considérations de système - y a t-il une limitation de la consommation du système en énergie ?
 - La vie en production - pendant combien de temps le processeur sera t-il maintenu en production ?
 - Coût - quel processeur répond le plus étroitement possible aux exigences de coût/performances de l'application ?

Choix du processeur ?

❑ Processeurs standards :

- Interface homme - machine ou réseau, couches applicatives hautes - Simplicité de mise en œuvre - flexibilité - Performances limitées

❑ DSP

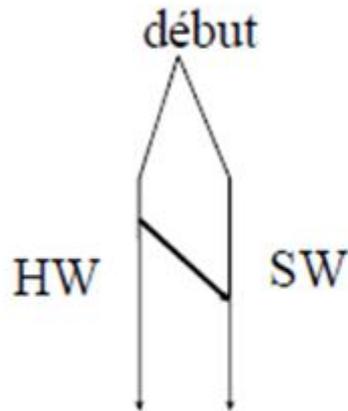
- Traitement du signal, opérations de calcul répétitives
 - **Note** : Les performances du processeur dépendent aussi
 - » Du compilateur
 - » De la mémoire cache (vitesse, taille, algorithme)
 - » De la bibliothèque de fonctions logicielles (écrites en assembleur)
 - » De la qualité de ses périphériques (communications, contrôle d'interruption, DMA, contrôleur mémoire externe, ...)

❑ Processeurs RISC-32 bits

- Il existe une multitude d'IP cores pour ces processeurs
 - Xilinx - MicroBlaze ; Altera - Nios ; IBM - PowerPC 440; Synopsys - MIPS4KEc ; ARM - ARM946E-S ; Tensilica - Xtensa ; LEON2 ; OpenRISC 1200.

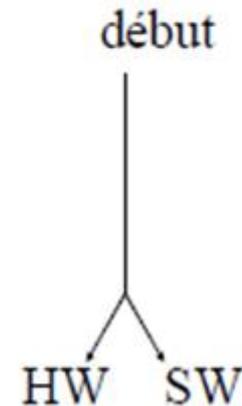
Conception et codesign

Conception traditionnelle



Réalisée par des groupes
d'ingénieurs indépendants

Codesign (flot concurrent)



Réalisée par le même groupe
d'ingénieurs en coopération

SE & Codesign

□ Approche traditionnelle

1. Choix du matériel (composants électroniques, processeur, ...) pour le système embarqué.
2. Donner le système ainsi conçu aux programmeurs.
3. Les programmeurs doivent réaliser un logiciel qui « colle » au matériel en n'exploitant que les ressources offertes.

Les systèmes embarqués sont de plus en plus complexes.

- Il est de plus en plus difficile de penser à une solution globale optimisée du premier jet.
- Il est de plus en plus difficile de corriger les « bugs ».
- Il est de plus en plus difficile de maintenir le système au cours du temps (obsolescence des composants, ...).



L'approche traditionnelle de développement d'un système embarqué doit évoluer ...

SE & Codesign

❑ Nouvelle approche

- ❑ Dans le processus de conception du système, on doit garder un niveau d'abstraction important le plus longtemps possible.
- ❑ Le système doit pouvoir être décomposé en sous-systèmes suivant une hiérarchie logique (approche objet).
 - Si le design change, on doit pouvoir en réutiliser une bonne partie (design reuse).
 - Il ne faut pas jeter à la poubelle un précédent design et repartir «from scratch» !

On doit pouvoir utiliser des outils de vérification automatique.

SE & Codesign

- ❑ Nouvelle approche (suite)
 - ❑ Quand on conçoit un système embarqué, un certain nombre de contraintes apparaissent :
 - Quel hardware a-t-on besoin ?
 - Quel processeur choisir ? Quelle puissance CPU ?
 - Quel type de mémoire et taille mémoire a-t-on besoin ?
 - Choix entre hardware rapide ou software intelligent ?
 - La consommation ? Minimiser les accès mémoire ? Choisir les instructions assembleur en fonction de leur consommation ?
 - Les contraintes de temps de conception seront-elles respectées ? Le TTM ?

SE & Codesign

- ❑ Nouvelle approche (suite)
 - ❑ Quand on conçoit un système embarqué, un certain nombre de contraintes apparaissent :
 - Est-ce que le système final marche correctement ?
 - Est-ce que les spécifications fonctionnelles ont été respectées ?
 - Comment doit-on tester les caractéristiques Temps Réel du système ?
 - Doit-on le tester avec des données réelles ?
 - Quelle plateforme de tests doit-on utiliser ?

SE & Codesign

- ❑ Nouvelle approche (suite)
 - ❑ Le *codesign* permet de concevoir à la fois le matériel et le logiciel pour une fonctionnalité à implémenter. Cela est maintenant possible avec les niveaux d'intégration offerts dans les circuits logiques programmables.
 - ❑ Le *codesign* permet de repousser le plus loin possible dans la conception du système les choix matériels à faire contrairement à l'approche classique où les choix matériels sont faits en premier lieu !

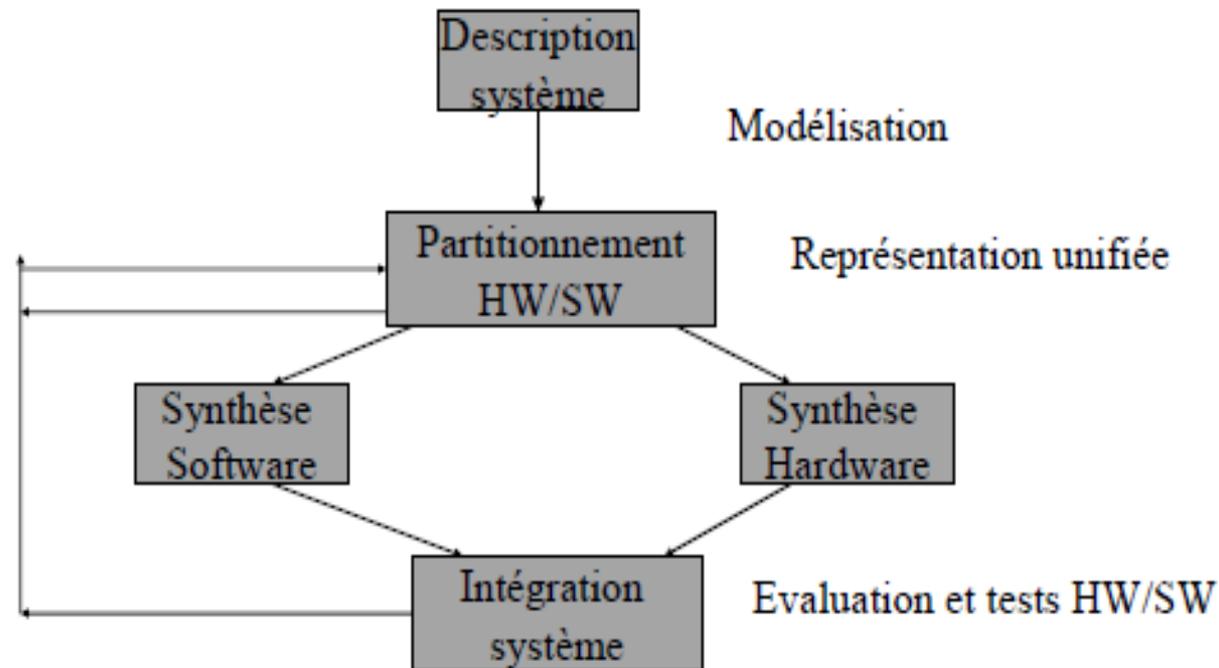
Etapes du Codesign

Nouvelle approche (suite)

Durant le processus de *codesign*, on distingue les étapes suivantes :

- Spécifications** : liste des fonctionnalités du système de façon abstraite.
- Modélisation** : conceptualisation et affinement des spécifications produisant un modèle du matériel et du logiciel.
- Partitionnement M/L.**
- Synthèse et optimisation** : synthèse matérielle et compilation logicielle.
- Validation** : cosimulation.
- Intégration.**
- Tests d'intégration.**

Etapes du Codesign



Avantages du codesign

- ❑ Le codesign est intéressant pour la conception de systèmes embarqués (ou de SoC) :
 - Amélioration des performances : parallélisme, algorithmes distribués, architecture spécialisée, ...
 - Reconfiguration statique ou dynamique en cours de fonctionnement.
 - Indépendance vis à vis des évolutions technologiques des circuits logiques programmables.
 - Mise à profit des améliorations des outils de conception fournis par les fabricants de circuits logiques programmables :
synthèse plus efficace, performance accrue.

Développement croisé

- ❑ L'environnement Intel x86 est loin d'être le plus utilisé dans les applications industrielles et embarquées (ex: PowerPC ou ARM sont parfois mieux adaptés).
- ❑ Cependant, la plupart des développeurs utilisent un PC intel (GNU/Linux ou Windows) comme poste de travail.
- ❑ Nécessité de mettre en place une **chaîne de développement croisée** permettant de produire du code non-intel sur un PC.

Compilation croisée

- ❑ La compilation croisée fait référence aux chaînes de compilation capables de traduire un code source en code objet dont l'architecture processeur cible diffère de celle où la compilation est effectuée.
- ❑ Produire du code pour un autre processeur

- ❑ **Compilation croisée ?**
 - ❑ Plus rapide sur la station de travail que sur le système cible
 - ❑ Plus facile d'avoir un environnement de développement sur la station de travail que sur la cible
 - ❑ Problème de place sur la cible
 - ❑ Problème de ressources sur la machine cible (écran, taille, clavier, souris, ...)

Bilan

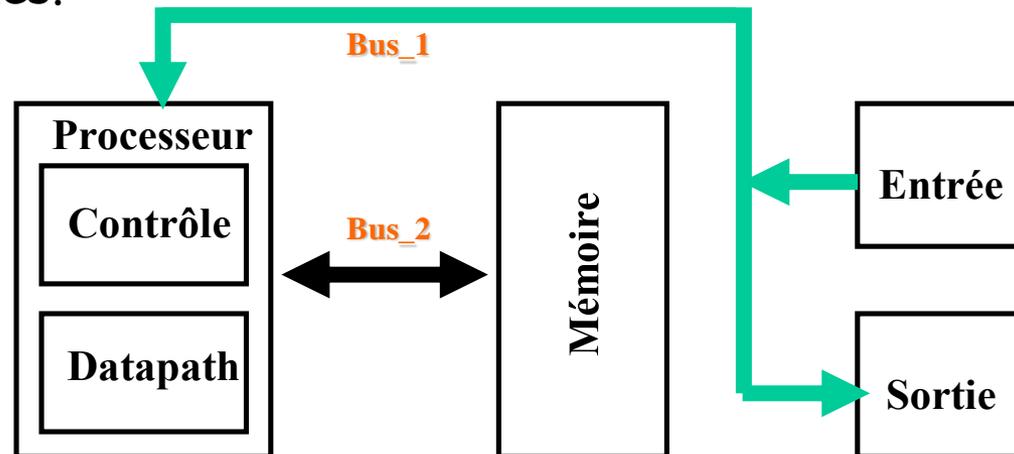
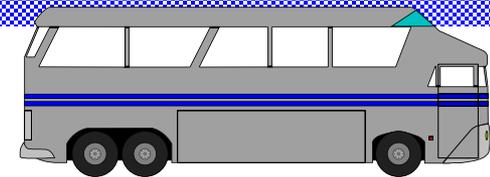
- ❑ Les systèmes embarqués étant de plus en plus complexes, une méthodologie rigoureuse de conception doit être maintenant mise en œuvre.
- ❑ Le codesign fait partie intégrante de cette méthodologie. Cela est maintenant possible grâce à des niveaux d'intégration sur silicium très importants des circuits logiques programmables qui peuvent aujourd'hui embarquer des processeurs.
- ❑ L'apparition de modules IP, véritables circuits électroniques sous forme logicielle *n'a fait qu'accentuer l'importance du codesign et du design reuse.*

Les Systèmes Embarqués (SE) et les systèmes sur puce (SoC) Bus

Définition d'un bus

Le Bus est :

- ❑ un lien de communication partagé
- ❑ un ensemble unique de fils (connexions) utilisé pour relier plusieurs sous-systèmes.



- ❑ Le Bus est aussi un moyen fondamental pour la composition de systèmes complexes

Exemples

- Un bus d'adresses sur 16 bits représente une capacité d'adressage maximale de :
 - 16^2
 - 2^{16}
 - 8^2
 - 16^8
- Les différents éléments d'un ordinateur mémoire, processeurs, ...) sont reliés par:
 - des registres
 - des pointeurs
 - le système d'exploitation
 - des bus
- La mémoire principale de MIPS R3000 est de 4Go. Quelle est la longueur du bus adresse ?
 - 30
 - 32^2
 - 32
 - 2^{32}

Conclusion

- ❑ Les systèmes embarqués
 - sont omniprésents à l'heure actuelle,
 - interagissent au sein de systèmes plus vastes
- ❑ Les challenges :
 - coût, performance, qualité
 - réglementations, responsabilités,
 - sûreté : disponibilité, fiabilité, sécurité-innocuité (non nuisible), sécurité-confidentialité,
 - maîtrise de leur développement, de leur évaluation
 - logiciels prépondérants + systèmes (environnement, matériels)

